



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,923	10/23/2003	Brendan McKeon	3382-66149	9047

26119 7590 11/25/2008
KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER

SHIH, HAOSHIAN

ART UNIT

PAPER NUMBER

2173

MAIL DATE

DELIVERY MODE

11/25/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/692,923

Applicant(s)

MCKEON ET AL.

Examiner

HAOSHIAN SHIH

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09/03/08.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9 and 21-40 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9 and 21-40 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/CDC)
- Paper No(s)/Mail Date _____

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Claims 1-9 and 21-40 are pending in this application and have been examined in response to application RCE filed on 09/03/2008.
2. The previously applied rejection under USC 112 is hereby withdrawn in view of applicant's amendment.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
4. Claim 1 recites the limitation "the computer" in last 2 lines of the claim. The Examiner suggests "**a** computer".
5. Claim 21 recites the limitation "the computer program" in last 2 lines of the claim. The Examiner suggests "**the source** computer".
6. Claim 28 recites the limitation "the computer program" in last line of the claim. The Examiner suggests "**the target** computer program".

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-9, and 21-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Armas and Mir Farooq ali et al. ("Mir", Building multi-platform user interfaces with UIML).

9. As to **INDEPENDENT** claim 1, De Armas discloses a method of generating identifier data for persistently identifying a user interface element of interest in a graphical user interface of a source computer program, the method comprising:

receiving data indicative of the user interface element of interest from a first software component (col.2, lines 59-65; a window object of interest is sent to a first software component navigator application); and

in response to receiving the data indicative of the user interface element of interest, generating an element path identifier of the user interface element of interest for persistently and uniquely identifying (col.3, lines 16-19; col.6, lines 15-20; attributes such as the parent/child/sibling relationships of each of the interface elements allows for the interface elements to be uniquely and persistently identified across different application states) the user interface element of interest and returning at least the unique element path identifier to the first software component (col.9, lines 27-31, 39-43; the element of interest is found via the element of interest's attributes, once the element of interest is found, a window handler that corresponds to the element of interest is returned to the navigator application);

wherein persistently and uniquely identifying the user element of interest comprises persistently and uniquely identifying the user interface element of interest across reboots of the computer running the source computer program (col.7, lines 9-12; element attributes are saved in a context data file to allow for other obvious usages to one of ordinary skill in the art, such as future retrievals). De Armas does not specifically disclose identifying the user interface element of interest across different states of the first software component.

In the same field of endeavor, Mir discloses identifying the user interface element of interest across different states of the first software component (sect 4.1, last par.; user interface elements are transformed to a specified software target platform or software state).

It would have been obvious to one of ordinary skill in the art, having the teaching of De Armas and Mir before him at the time the invention was made, to modify the combined attributes usable to strongly define user interface elements taught by De Armas to include a user interface platform identifier taught by Mir with the motivation being to allow a unified user interface element identifier description across different platforms for the purpose of reduce redundancy in software development (Mir, sect. 1 "Introduction", par. 3).

10. As to claim 2, De Armas discloses wherein generating the element path identifier is implemented by a second software component communicative with the source computer program (col.5, lines 30-34; a software analysis tool is used to form a hierarchical tree of the source computer program).

11. As to claim 3, De Armas discloses wherein generating an element path identifier comprises:

using a hierarchical tree structure representation a function description of the graphical user interface to locate a leaf node related to the user interface element of interest (col.5, lines 1-2; a hierarchical representation of elements in an application is presented; fig.1B; the functional behavior of object "28" is to frame objects "30" and "32");

storing exposed identifier information of the user interface element of interest in an element path identifier data structure (col.3, lines 22-29; various attributes relating to the user interface elements are stored); and
proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes of the user interface element of interest in the element path identifier data structure (col.5, lines 60-64; parent/child relationship between window objects are preserved).

12. As to claim 4, De Armas discloses converting the element path identifier to a string type data structure (col.3, lines 22-27).

13. As to claim 5, De Armas discloses receiving data indicative of a designated element path root node of the tree structure representation, wherein the step of proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes is continued only until the element path root node is reached (col.5, lines 2-6; a complete analysis of the source computer program is performed to indicate parent/child relations and various other attributes associated with interface elements of the source program).

14. As to claim 6, De Armas discloses wherein the exposed identifier information is a local alpha numeric identifier not guaranteed to be unique, a class name, a module name associated with an application program or a sibling order (col.7, lines 42-43; objects at their respective local level can share a common set of attributes).

15. As to claim 7, De Armas discloses determining that at least one of the selected parent nodes is a root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier (col.5, lines 44-46, 52-55; a sub-context tree is created for each application state, since the parent/child relations are preserved in each of the sub-context tree representing each of

the application states, there must be at least one unique child element from each sub-context tree because each application state is unique from each other) and

storing in the element path identifier data structure, the named branch element identifier for the at least one user interface element within the scope of the named branch portion (fig.5A).

16. As to claim 8, De Armas discloses storing in the element path identifier a strong name associated with the strongly named branch along with the named branch element identifier (col.5, line 45; a child of a strongly named branch parent encompasses a strong name since information from the parent is contained in the child).

17. As to claim 9, De Armas discloses wherein the named branch element identifier stored in the element path identifier data structure is for the user interface element of interest and no identifying information related to the parent elements within the scope are stored in the element path identifier data structure (col.7, lines 42-43; objects at their respective local level can share a common set of attributes).

18. As to **INDEPENDENT** claim 21, De Armas discloses at least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component, the function comprising:

an input parameter representing a user interface element of interest in a graphical user interface of a source computer program (col.2, lines 59-65; a window object of interest is sent to a first software component navigator application);

an output parameter representing an element path identifier for persistent unique identification of the user interface element of interest; wherein the element path identifier comprises a hierarchical path of the inheritance from the user interface element of interest to a parent root element (col.5, line 45; col.6, lines 15-20; attributes such as the parent/child/sibling relationships of each of the interface elements allows for the interface elements to be uniquely and persistently identified); and executable software for receiving the input parameter representing a user interface element of interest and in response, generating the output parameter representing an element path identifier of the user interface element of interest (col.9, lines 27-31, 39-43; the element of interest is found via the element of interest's attributes, once the element of interest is found, a window handler that corresponds to the element of interest is returned to the navigator application) such that the output parameter represents an identifier capable of persistently identifying the user interface element of interest (col.3, lines 16-19; col.6, lines 15-20; attributes such as the parent/child/sibling relationships of each of the interface elements allows for the interface elements to be uniquely and persistently identified across different program states or builds). De Armas does not specifically disclose identifying the user interface element of interest across different builds [and] across multiple states of the computer program.

In the same field of endeavor, Mir discloses identifying the user interface element of interest across different builds [and] across multiple states of the computer program (sect 4.1, last par.; user interface elements are transformed to a specified software target build).

It would have been obvious to one of ordinary skill in the art, having the teaching of De Armas and Mir before him at the time the invention was made, to modify the combined attributes usable to strongly define user interface elements taught by De Armas to include a user interface platform identifier taught by Mir with the motivation being to allow a unified user interface element identifier description across different platforms for the purpose of reduce redundancy in software development (Mir, sect. 1 "Introduction", par. 3).

19. As to claim 22, De Armas discloses wherein the element path identifier is a string type data structure (col.3, lines 22-27).

20. As to claim 23, De Armas discloses wherein the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest (col.5, lines 18-21; "window objects").

21. As to claim 24, De Armas discloses wherein the element path identifier comprises class names of component elements of an element path of the element of interest (col.3, lines 22-25; "class name").
22. As to claim 25, De Armas discloses wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest (col.5, lines 10-15; "top-level window object").
23. As to claim 26, De Armas discloses wherein the element path identifier comprises sibling order data (col.6, lines 17-20; "siblings").
24. As to claim 27, see rationale addressed in the rejection of claim 7 above.
25. As to **INDEPENDENT** claim 28, see rationale addressed in the rejection of claim 21 above.
26. As to claim 29, see rationale addressed in the rejection of claim 22 above.
27. As to claim 30, see rationale addressed in the rejection of claim 23 above.
28. As to claim 31, see rationale addressed in the rejection of claim 24 above.

29. As to claim 32, see rationale addressed in the rejection of claim 25 above.
30. As to claim 33, see rationale addressed in the rejection of claim 26 above.
31. As to claim 34, see rationale addressed in the rejection of claim 7 above.
32. As to **INDEPENDENT** claim 35, see rationale addressed in the rejection of claim 21 above.
33. As to claim 36, De Armas discloses persistently uniquely identifying the user interface element of interest across different builds of the source computer program (col.3, lines 16-19; col.6, lines 15-20; attributes such as the parent/child/sibling relationships of each of the interface elements allows for the interface elements to be uniquely and persistently identified). De Armas does not specifically disclose identifying the user interface element of interest across different builds of the source computer program.

In the same field of endeavor, Mir discloses identifying the user interface element of interest across different builds of the computer program (sect 4.1, last par.; user interface elements are transformed to a specified software target build).

34. As to claim 37, see rationale addressed in the rejection of claim 36 above.

35. As to claim 38, De Armas discloses the element path identifier identifies the source computer program in which the user interface element resides (col.3, lines 15-20; "target application").

36. As to claim 39, De Armas discloses the element path identifier identifies a plurality of element attributes (col.3, lines 22-23). De Armas does not disclose an identifier that identifies a user interface platform used to create the user interface element.

In the same field of endeavor, Mir discloses an identifier that identifies a user interface platform used to create the user interface element (sect 4.1, last par.).

37. As to claim 40, De Armas discloses the element path identifier identifies an application frame in which the user interface element resides (fig.1B; parent window frame "26" and child window frame "28").

Response to Arguments

38. Applicant's arguments filed 09/03/2008 have been fully considered but they are not persuasive.

39. Applicant argues that De Armas does not disclose the functional ancestor elements and sibling elements of the graphical user interface.

In response to applicant's argument, De Armas discloses functional ancestor elements such as the window element "26" that hosts child and sibling elements such as a list box "30" that lists items and an edit box "32" that allows for editing (fig.1B; col.5, lines 9-20). Further, each of the elements are associated with a vocabulary to allow for vocal commands of the element of interest (col.8, lines 33-39).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HAOSHIAN SHIH whose telephone number is (571)270-1257. The examiner can normally be reached on m-f 0730-1700.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dennis Chow can be reached on (571) 272-7767. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

HSS

/DENNIS-DOON CHOW/
Supervisory Patent Examiner, Art Unit 2173